



# **Programma PRIN**

**(gruppo Chieti-Pescara: Sciarra, Pasculli,  
Calista, Vessia, Santoro)**

14 dicembre 2012

## Modellazione matematico-numericò dell'instabilità di versanti granulari, dall'innescò allo sviluppo dell'eventuale debris flow

- *Le fenomenologie che il gruppo propone di studiare riguardano i processi d'innescò d'instabilità di ammassi granulari, a causa sostanzialmente di eventi piovosi, e quindi i relativi cedimenti fino allo sviluppo di eventuali flussi rapidi.*
- *A tale scopo, saranno sviluppati modelli matematico-numericò 2D (eventualmente 3D), implementati ed elaborati mediante l'ultima generazione del Compilatore Fortran (e/o C, C++). In maniera innovativa, sarà esplorato anche l'uso del calcolo parallelo.*

## Aspetti fenomenologici

- **caratterizzazione delle eterogeneità e asimmetrie di ammassi granulari**
  - ✓ *porosità variabile anche per il carico litologico;*
  - ✓ *presenza di debolezze interne;*
  - ✓ *stratificazioni, lenti, faglie;*
  - ✓ *connettività tra pori.*
  
- **infiltrazione bifasica (aria e acqua) in condizioni insature e/o sature**
  - ✓ *eventuale ruscellamento;*
  - ✓ *escursione termica giornaliera e stagionale.*
  
- **instabilità**
  - ✓ *piccoli cedimenti;*
  - ✓ *grandi deformazioni;*
  - ✓ *flussi veloci (comportamento 'fluid like' e 'solid like' dei diversi costituenti il materiale.*

## Modelli matematici

- **caratterizzazione delle eterogeneità e asimmetrie di ammassi granulari:**
  - ✓ *approcci stocastici e frattali*
- **infiltrazione bifasica (aria e acqua) in condizioni insature e/o sature:**
  - ✓ equazione di **Richards** con il termine dello “swelling”; “shift” di variabile;
  - ✓ equazioni di **Nithiarasu** e legge di **Forkheimer** per la **variabilità continua** della porosità e “dual porosity” per la **variabilità discontinua**;
  - ✓ variabilità con la temperatura della viscosità dell’aria e dell’acqua nel suolo.
- **Instabilità:**
  - ✓ equazioni costitutive visco-plastiche;
  - ✓ equazioni non stazionarie di **Cauchy**;
  - ✓ equazioni mediate di **Navier Stokes** relative alla **turbolenza** dei muddy-debris flow (*k*-Omega, LES);
  - ✓ equazioni costitutive per fluidi **non Newtoniani** di **Herschel-Bulkley**.

- **elaborazioni stocastiche:**
  - ✓ utilizzo processi di elaborazioni numeri pseudo-random ed applicazione anche di leggi proposte dagli autori;
- **infiltrazione, instabilità, muddy-debris flow:**
  - ✓ approcci Euleriani e Lagrangiani;
  - ✓ approcci mesh-free puri (Smoothed Particle Hydrodynamics **SPH**, **Monaghan, Morris, Liu**);
  - ✓ approcci mesh-free ibridi (Particle Finite Element Method **PFEM**, **Oñate**)
- **tecniche numeriche e di programmazione:**
  - ✓ aggiornamento della griglia numerica mediante l'approccio del 'Constraint Delaunay' allo scopo di includere eventuali faglie;
  - ✓ soluzioni di equazioni differenziali non lineari (tipo Newton-Rapshon e varianti);
  - ✓ soluzione delle Equazioni mediate di Navier Stokes con criteri di stabilizzazione per flussi 'convective-like'.
  - ✓ programmazione in Fortran 2010 e C++ con esplorazione del GPU e CUDA (calc. Parallelo)

## Cronologia (nove quadrimestri)

### ▪ **Anno I**

- ✓ 1) griglia numerica flessibile (Delaunay);
- ✓ 2) testing del punto 1) doubly porosity e swelling;
- ✓ 3) infiltrazione;

### ▪ **Anno II**

- ✓ 4) testing del modello di cui al punto 3);
- ✓ 5) introduzione delle equazioni costitutive e testing;
- ✓ 6) accoppiamento infiltrazione e stato tensionale

### ▪ **Anno III**

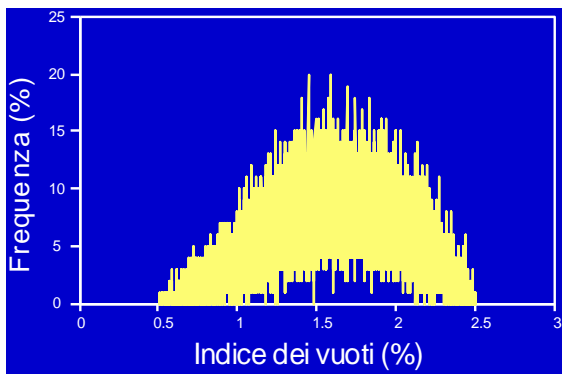
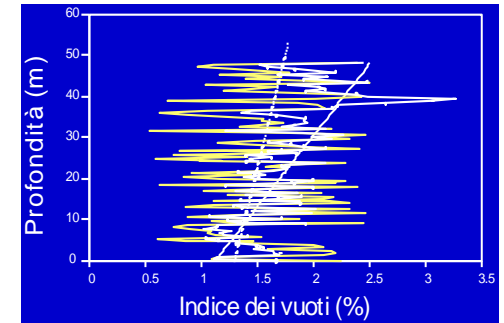
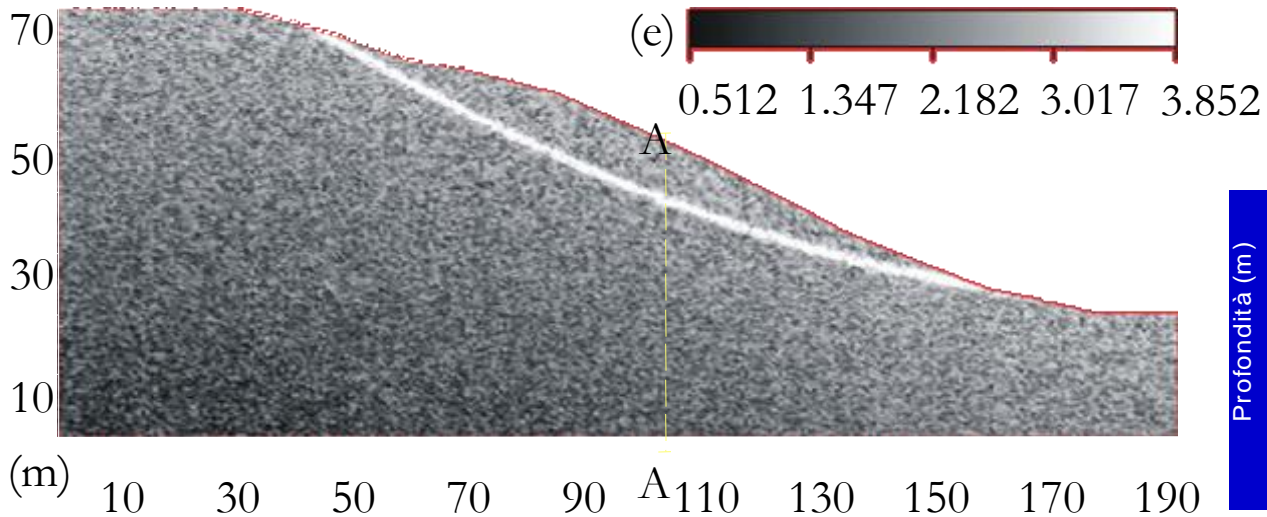
- ✓ 7) criteri di rottura e propagazione delle micro rotture;
- ✓ 8) accoppiamento ed ulteriore sviluppo modello numerico di muddy-debris flow;
- ✓ 9) approfondimenti con le altre UR.

### ▪ **Anni I – II - III**

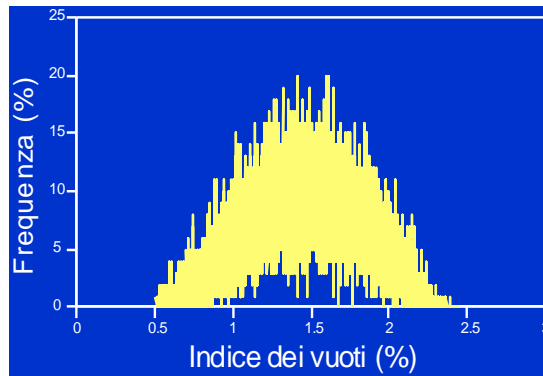
- ✓ elaborazioni numeriche mediante codici commerciali di instabilità di versanti reali;
- ✓ collaborazioni con altre UR con simile tipologia di attività di ricerca.

Pasculli & Sciarra (2003)

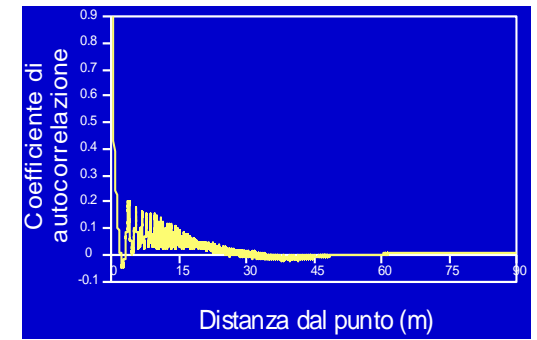
## Analisi agli Elementi Finiti per l'analisi delle eterogeneità multiscala



$C1=C2=1$

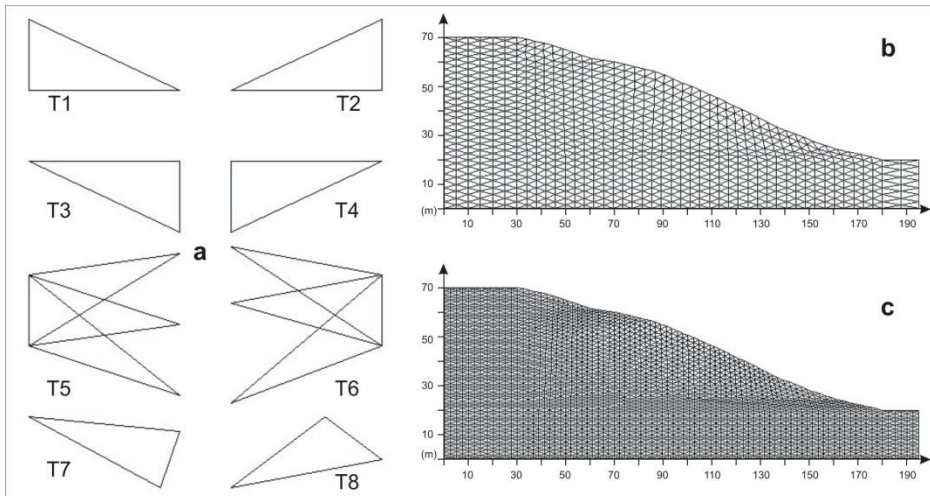


$C1=1$   $C2=0.8$



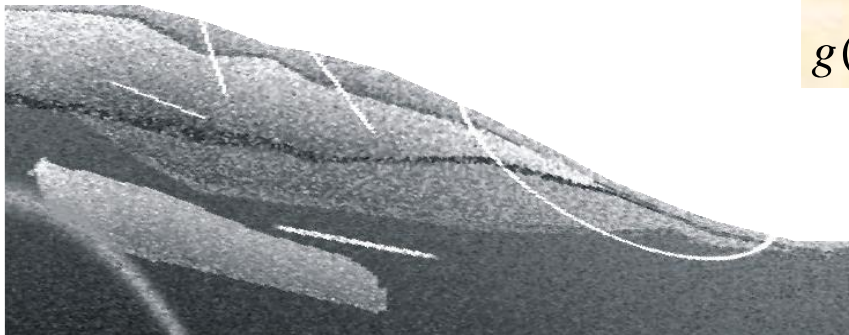
# Esperienza acquisita dall'Unità di Ricerca

Pasculli & Sciarra (2003)



$$\begin{aligned} Phys(x_g, y_g) = & Phys\_rand(x_g, y_g) + \\ & + [\mu(x_g, y_g) - Phys\_rand(x_g, y_g)] \cdot [mean + \\ & + \sigma \cdot G\_norm] \cdot e^{-\left[\frac{f(x_g, y_g)}{s}\right]^2} \end{aligned}$$

$$g(p^2) = e^{-\left[\frac{f(x_g, y_g)}{s}\right]^2}$$

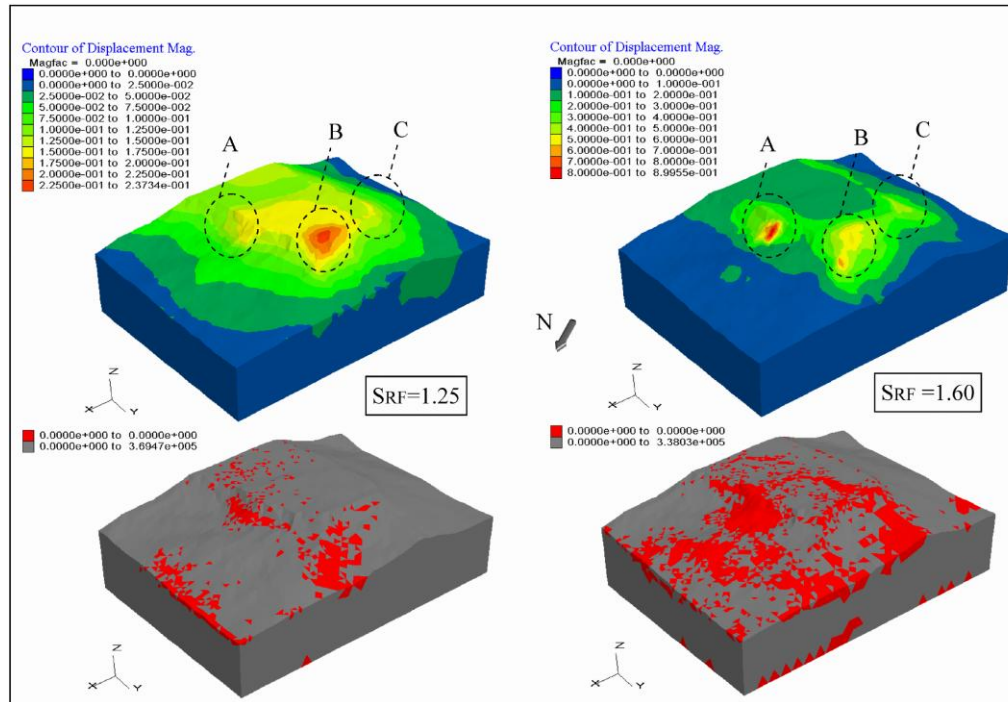




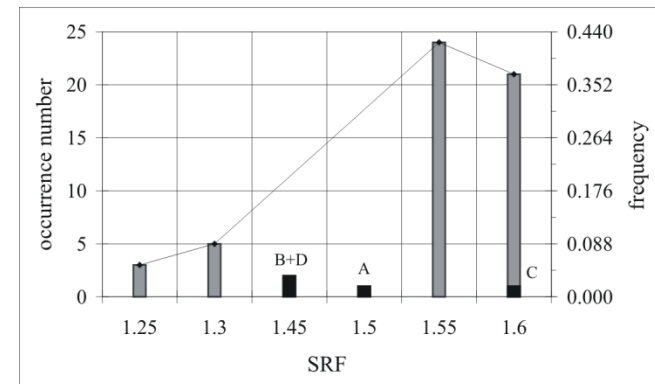
# Esperienza acquisita dall'Unità di Ricerca

The effects of spatial variability of mechanical parameters on a 3D landslide study

A. Pasculli, N. Sciarra, M. Calista & M. Mangifesta (2006)



Analisi 3D dell'instabilità di versanti a parametri meccanici costanti, confrontate con i risultati conseguiti mediante Approccio Probabilistico, includendo eterogeneità a diverse scale spaziali



## A 2D NUMERICAL MODELLING OF WATERY DEBRIS FLOW INCLUDING A STATISTICAL DETERMINATION OF THE LOCAL EROSION Pasculli & Sciarra (2007)

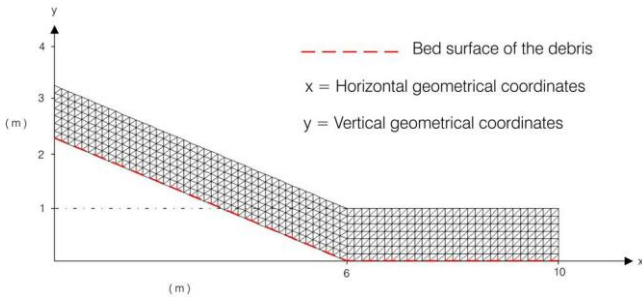
### Metodo agli elementi Finiti

$$\begin{cases} \frac{\partial(\rho_f U_{fx})}{\partial t} + \frac{\partial(\rho_f U_{fx} U_{fx})}{\partial x} + \frac{\partial(\rho_f U_{fx} U_{fy})}{\partial y} = -\frac{\partial P}{\partial x} + \rho_f g_x + \frac{\partial}{\partial y} \left[ (\mu_r + K_M) \frac{\partial U_{fx}}{\partial y} \right] \\ \frac{\partial(\rho_f U_{fy})}{\partial t} + \frac{\partial(\rho_f U_{fy} U_{fx})}{\partial x} + \frac{\partial(\rho_f U_{fy} U_{fy})}{\partial y} = -\frac{\partial P}{\partial y} + \rho_f g_y + \frac{\partial}{\partial y} \left[ (\mu_r + K_M) \frac{\partial U_{fy}}{\partial y} \right] \\ \frac{\partial U_{fx}}{\partial x} + \frac{\partial U_{fy}}{\partial y} = 0 \end{cases}$$

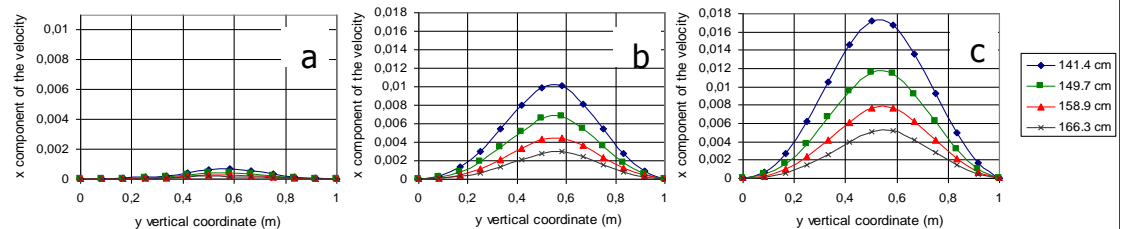
$$\bar{\mathbf{P}}^{n+1} = - \left[ \hat{\mathbf{L}}^n + \frac{\Delta t}{\rho} \cdot \mathbf{L} \right]^{-1} \left[ \mathbf{G}^T \cdot \tilde{\mathbf{U}}^{n+1} - \alpha \cdot \frac{\Delta t}{\rho} \cdot \mathbf{L} \cdot \bar{\mathbf{P}}^n + \mathbf{Q} \cdot \bar{\boldsymbol{\pi}}^n \right]$$

$$\bar{\mathbf{U}}^{n+1} = \tilde{\mathbf{U}}^{n+1} + \Delta t \cdot \mathbf{M}_d^{-1} \cdot \mathbf{G} \cdot (\bar{\mathbf{P}}^{n+1} - \alpha \cdot \bar{\mathbf{P}}^n)$$

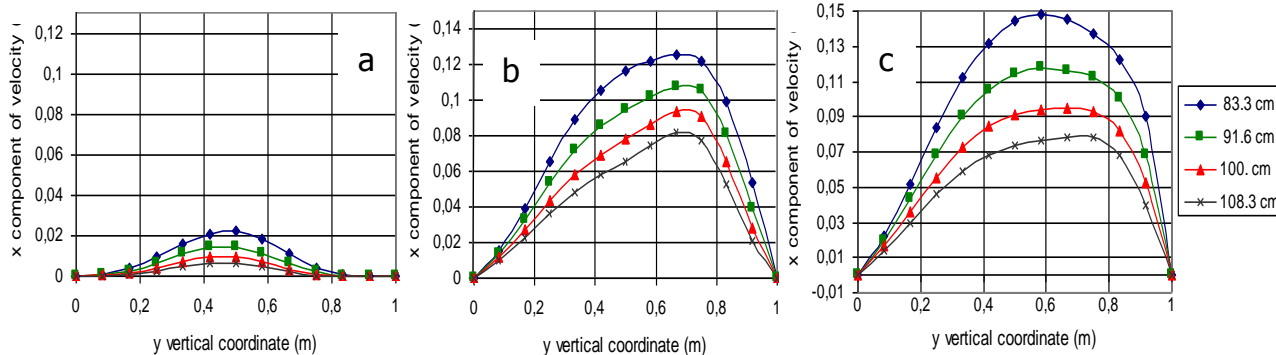
$$\bar{\mathbf{c}}^{n+1} = -\mathbf{M}_d^{-1} \cdot \hat{\mathbf{C}}^{n+1} \cdot \bar{\mathbf{U}}^{n+1} \quad \bar{\boldsymbol{\pi}}^{n+1} = -\hat{\mathbf{M}}_d^{-1} \cdot \mathbf{Q}^T \cdot \bar{\mathbf{P}}^{n+1}$$



densità uniforme del debris flow



densità asimmetrica



# Esperienza acquisita dall'Unità di Ricerca

(subroutine approccio Delaunay per generazione automatica mesh sviluppata dagli autori 2011)

```
SUBROUTINE DELAUNAY
-----
!-----
USE DFLIB      !Intrinsic Compiler Graphic
Routines
USE MOD_PARAM  !Array parameters
USE MOD_REAL   !Real variables
USE MOD_IND    !Index parameters and
integer variables
USE MOD_CHARACTER !Characters
(writeable only if they are read)
C USE MOD_LOGIC !Logical variables
!-----
!-----
REAL(8) :: EX1,EX2,EX3,EY1,EY2,EY3
REAL(8) :: XMIN,XMAX,YMIN,YMAX
REAL(8) :: AX1,AX2,AX3,AY1,AY2,AY3
!-----
!=====
!----- Start of Nodes Ordering -----
!-----
!-After this call, V_XORD(I) and V_YORD(I)
are ordinated
! by growing values of both XI(I) and YI(I)
!-In entrance are requested only V_XORD()
and V_YORD() and
! the initialization by loop_1_1
!-For each time-step a XG() and YG()
nodes ordering is necessary
!-----
C IF(NEL_TOT.GE.30) THEN
C   I_ST=NEL_TOT-29
C   ELSE
C   I_ST=1
C   END IF
!-----
DO I=1,NOD_IN      !-----start loop_1_1-----
-----
I_IND(I) = I
K_OLD(I) = I
V_XORD(I) = XG(I)
V_YORD(I) = YG(I)
VARIABLE(I)= XG(I)
END DO      !-----end loop_1_1----
-----
NUMBER=NOD_IN
CALL ORD_NODES(NUMBER)
```

```
!-----
CALL CPU TIME (TIME_ORD_NODES)
!Mesaures CPU T0 time
WRITE(*,*) 'TIME_ORD_NODES
=,TIME_ORD_NODES-T_P(2)
T_P(3)=TIME_ORD_NODES-T_P(2)
!----- End of Nodes First ordering -----
!=====
!----- Routine to search an embedding rectangle
of -----
!-----the whole set of points -----
CALL EMBED_RECTANGL
!-----
CALL CPU TIME (TIME_EMBED_RECTANGL)
!Mesaures CPU T0 time
WRITE(*,*) 'TIME_EMBED_RECTANGL
=,TIME_EMBED_RECTANGL-T_P(3)
T_P(4)=TIME_EMBED_RECTANGL-T_P(3)
!-----
!=====
N_ELEM(1)=4 !Initialization is performed in
INITIAL
NEL_TOT =N_ELEM(1)
!===== Start elements numbering and
construction =====
!-----anticlock-wise of nodes for each
element-----
!-----
DO IE=1,NEL_TOT !-----start loop_2---
NOD_EL(IE)=3 !Number of nodes belonging to
element IE
changed !It is the first guess; it may be
changed
!-----
ELEM_INSERT lin
END DO !-----end loop_2---
!-----
!IMI allows to introduce just one node
each time
IMI=0
DO K=1,NOD_IN-IMI !-----start loop_1_5-----
!-----
IF(K.EQ.1) THEN
CALL FIRST_NODE
CALL ELEM_GEOM
```

```
!-----
CALL CPU TIME (TIME_FIRST_NODE)
!Mesaures CPU T0 time
WRITE(*,*) 'TIME_FIRST_NODE
=,TIME_FIRST_NODE-T_P(4)
T_P(5)=TIME_FIRST_NODE-T_P(4)
!-----
CC CALL ELEM_GEOM
! the routine is called inside CIRCUMCENTER
!-----
!-----
CALL CPU TIME (TIME_ELEM_GEOM)
!Mesaures CPU T0 time
WRITE(*,*) 'TIME_ELEM_GEOM
=,TIME_ELEM_GEOM-T_P(5)
T_P(6)=TIME_FIRST_NODE-T_P(5)
!-----
ELSE
!-----
!-Number of the elements associated to the
! insertion of the node K. Initially
N_ELEM(1)
! is indicated in INITIAL. Then it is updated
! in ELEM_INSERT
!-----
!-----
!----- Elements First ordering -----
!-----
GO TO 3000
WRITE(9,*) ' from DELAUNAY '
WRITE(9,*) '----- elements ordering by
ORD_ELEMS(NUM) -----'
DO IE=1,NEL_TOT
I_IEL(IE) =IE
V_XORD(IE) =XCC(IE)
V_YORD(IE) =YCC(IE)
END DO
!-----
CALL ORD_ELEMS(NEL_TOT)
!-----
!-----Updated initial nodes coordinates -
WRITE(9,*) ' IE IEL_OLD(IE) V_XORD(IE)
V_YORD(IE)
$ V_RCC(IE)'
!-----
DO IE=1,NEL_TOT
XCC(IE)=V_XORD(IE)
YCC(IE)=V_YORD(IE)
RCC(IE)=V_RCC(IE)
```

# Esperienza acquisita dall'Unità di Ricerca

(CONTINUE, subroutine approccio Delaunay per generazione automatica mesh sviluppata dagli autori 2011)

```

WRITE(9,70)
IE,IEL_OLD(IE),V_XORD(IE),V_YORD(IE),V_RCC(IE)
END DO
!-----
70 FORMAT(2I6,3(4X,F12.6))
WRITE(9,*) '
!----- End of elements First ordering-----
3000 CONTINUE

!=====
NOD_DYN=4+K+1
CALL NOD_INSERT(K,NOD_DYN)
!In this routine is updated also the number
of elements due to
!the insertion of a new node (k+1). Thus
new elements are built
!and some of the previous elements are
deleted
!=====
!=====
!=====
!some elements are eliminated and new
ones are introduced
!the new value of elements is calculated in
ELEM_INSERT
CALL ELEM_INSERT(K,NOD_DYN)
!-----
NEL_TOT=N_ELEM(K) !updating the number of
elements

!=====
!=====
!nodes and elements links
C IF(K.GE.3) THEN
C CALL NOD_ELEM_LINK
C ELSE
C END IF
!=====
C WRITE(*,*) ' IE K XL(IE,3) YL(IE,3)
C $ X_GLB(K) Y_GLB(K)'
C DO IE=1,N_ELEM(K)
C WRITE(*,80)
IE,K,XL(IE,3),YL(IE,3),X_GLB(K),Y_GLB(K)
C END DO
C 80 FORMAT(2I6,4(F12.6))
C PAUSE

!-----
!-----The loop recognize the type of each
element
!--UNIT=12 is MESH_DATA-----
CW WRITE(12,*) ' from DELAUNAY '
CW WRITE(12,*) ' K IE N_ELEM(K) IFLAG(IE)
IN ID_IE(IE)
CW $ XL(IE,IN) YL(IE,IN)'

DO IE=1,N_ELEM(K) !In this loop, each
element is described by nodes
!coordinates I_EL
IEPER =1
IESUM =0
DO IN=1,NOD_EL(IE) !IN is the local numeration,
while
!NOD_EL(IE) is the maximum
node number for each element
!K is the global number of each node
IF((XL(IE,IN).NE.XB_MINER).AND.(XL(IE,IN).NE.XB_MA
XER)) THEN
IB =1
ELSE
IB =0
END IF
IEPER=IEPER*IB
IESUM=IESUM+IB
END DO
ID_IE(IE)=IEPER
!!D_IE(IE)=IESUM

!-----
!---- UNIT=12,FILE='MESH_DATA.TXT'-----
CW DO IN=1,NOD_EL(IE)
CW WRITE(12,20)
K,IE,N_ELEM(K),IFLAG(IE),IN,ID_IE(IE),
CW $ XL(IE,IN),YL(IE,IN)
CW END DO
!-----
CW WRITE(12,*) '
END DO

!-----
20 FORMAT(2I6,4(3X,I6),4(4X,F10.6))

!=====
!-----geometrical triangles parameters -----
CALL ELEM_GEOM
!inside CIRCUMCENTER is called
CW WRITE(9,*) ' END OF INSERTION OF
NODE', K=',K
CW WRITE(9,*) '

!=====
END IF
END DO !-----End loop_1_5 (do
k=1,n)
!-----
!-----
CALL CPU_TIME (TIME_EL_NOD_INS) !Mesaures
CPU T0 time
WRITE(*,*) 'TIME_EL_NOD_INS
=',TIME_EL_NOD_INS-T_P(6)
T_P(7)=TIME_EL_NOD_INS-T_P(6)
!-----
C WRITE(*,*) NOD_IN
C PAUSE
CALL MESH_FINAL
C WRITE(*,*) NOD_IN
C PAUSE
CALL CPU_TIME (TIME_MESH_FINAL) !Mesaures
CPU T0 time
WRITE(*,*) 'TIME_MESH_FINAL
=',TIME_MESH_FINAL-T_P(7)
T_P(7)=TIME_MESH_FINAL-T_P(7)
CALL CPU_TIME (TIME_IN_1)
WRITE(*,*) 'T1 =',TIME_IN_1
DT_INDAT=TIME_IN_1-TIME_IN_0
WRITE(*,*) 'DT: T1-T0=',DT_INDAT
WRITE(*,*) '
PAUSE
CP CALL GRAPH_BASE
!-----
END SUBROUTINE DELAUNAY

```

# Esperienza acquisita dall'Unità di Ricerca (Simulazione numerica SPH di un muddy debris flow effettuato in laboratorio) (submitted 2012)

